Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: https://orca.cardiff.ac.uk/id/eprint/101465/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Shao, Jianhua ORCID: https://orcid.org/0000-0001-8461-1471 and Ong, Hoang 2017. Exploiting contextual information in attacking set-generalized transactions. ACM Transactions on Internet Technology 17 (4) , 40. 10.1145/3106165 file

Publishers page: https://doi.org/10.1145/3106165 <https://doi.org/10.1145/3106165>

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

information services
gwasanaethau gwybodaeth

A

Exploiting Contextual Information in Attacking Set-Generalized Transactions

JIANHUA SHAO and HOANG ONG, Cardiff University

Transactions are records that contain a set of items about individuals. For example, items browsed by a customer when shopping online form a transaction. Today, many activities are carried out on the Internet, resulting in a large amount of transaction data being collected. Such data are often shared and analyzed to improve business and services, but they also contain private information about individuals that must be protected. Techniques have been proposed to sanitize transaction data before their release, and set-based generalization is one such method. In this paper, we study how well set-based generalization can protect transactions. We propose methods to attack set-generalized transactions by exploiting contextual information that is available within the released data. Our results show that set-based generalization may not provide adequate protection for transactions, and up to 70% of the items added into the transactions during generalization to obfuscate original data can be detected by our methods with a precision over 80%.

Categories and Subject Descriptors: H.2.7 [**Database Administration**]: Security; Protection

General Terms: Design; Performance

Additional Key Words and Phrases: Privacy, de-anonymization, transaction data, semantic relationship

ACM Reference Format:
Jianhua Shao and Hoang Ong. 2015. Exploiting Contextual Information in Attacking Set-Generalized Transactions. *ACM Trans. Internet Technol.* V, N, Article A (January YYYY), 20 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Transactions are records that contain a set of items about individuals, and they are being increasingly collected, shared and analyzed as a result of increased activities on the Internet and widespread deployment of data capturing applications. For example, when patients visit a hospital, their diagnostic data may be recorded and then used in medical studies, and when customers shop online, their browsing activities may be retained by the vendor to help recommend products to other customers. Such data are valuable to the society and organizations as their analysis can help improve business intelligence and healthcare provision.

However, transaction data may contain personal and sensitive information, and publishing them directly can risk privacy breaches [Golle 2006; Narayanan and Shmatikov 2008]. Unfortunately, de-identification (i.e. removing personal identifiers) may not provide sufficient protection for individuals' privacy [Barbaro and Zeller 2006]. For example, Fig. 1 shows a set of 4 transactions, each recording some medical conditions associated with a patient. If an adversary knows that Mary has a *blood pressure* condition and she is in the dataset, he or she can infer that transaction 1 belongs to Mary, and find out other information about her.

Author's addresses: J. Shao and H. Ong, School of Computer Science & Informatics, Cardiff University, Cardiff, CF24 3AA, UK.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© YYYY ACM 1533-5399/YYYY/01-ARTA $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

TID	Items
1	heart disease, blood pressure, icd, weakness, dizziness
2	anesthesia, icd, pain, diabetes
3	gangrene, limbs, injury
4	knee, injury

Fig. 1: An example of transaction data

To protect transactions against potential privacy disclosure, techniques have been developed to anonymize them. One such technique is set-based generalization which attempts to hide an original item by replacing it with a set of items. For example, Fig. 2 is a set-generalized version of Fig. 1 where *blood pressure*, for instance, is generalized to (*blood pressure, icd, limbs, injury*). As such, knowing that Mary has a *blood pressure* condition will no longer be enough to link Mary to transaction 1 with certainty.

TID	Items
1	heart disease, (*blood pressure, icd, limbs, injury*), weakness, dizziness
2	anesthesia, (*blood pressure, icd, limbs, injury*), pain, diabetes
3	gangrene, (*blood pressure, icd, limbs, injury*)
4	knee, (*blood pressure, icd, limbs, injury*)

Fig. 2: Set-generalized transactions

It has been shown that set-based generalization is more flexible and can retain data utility better than other generalization methods [Loukides et al. 2013]. Data publishers can specify which items to protect (e.g. *blood pressure* and *icd* in Fig. 2) and to what extent (e.g. having at least 4 transactions to cover them). However, set-based generalization does not consider the transaction as a whole when forming a set to generalize an item. This makes it vulnerable to attacks that use other items in the transaction as a context. For example, consider the generalized transaction 4 in Fig. 2. Although it suggests that *blood pressure*, *icd*, *limbs*, *injury* or any combination of them are possible, the presence of *knee* hints that it is more likely to be *injury* than *icd*. This type of semantic analysis will allow an adversary to reduce a generalized item to its original form, thereby breaking the protection for the data.

The first attempt to reconstruct original transactions from their set-generalized versions through semantic analysis was reported in [Ong and Shao 2014]. Their methods use Normalized Google Distance (NGD) [Cilibrasi and Vitányi 2007] to score semantic relationships between items, and eliminate any item in a generalized item (e.g. *blood pressure*) that is deemed to have a weak relationship with a contextual item (e.g. *knee*) based on their NGD score. However, these methods are threshold-based and can lead to wrong eliminations when a rare item occurs in a transaction. This is because a rare item tends to have a higher NGD than a more common item does when assessing their semantic relationships with another item, even if the rare item is more strongly related to the given item than the more common one is.

In this paper, we address this issue. We propose new methods to attack set-generalized transactions. More specifically, we make the following contributions:

— Similar to the methods proposed in [Ong and Shao 2014], we build our new methods on NGD. This measure establishes semantic relationship between two terms by querying the Google repository of WWW pages: the more pages in which the two terms appear together, the more related they are considered to be. This allows both

term co-occurrence and *semantic similarity* to be measured, as we shall explain later, and eliminates the need to construct a comprehensive dictionary or a corpus for testing item relationships. This makes our approach generic and practical.

— We propose two clustering-based methods that aim to identify comparable NGD distances first, then apply heuristics to eliminate items from a cluster that is deemed to be more likely to contain an item that is not original but is added to a transaction during generalization. This is in contrast to the methods proposed in [Ong and Shao 2014] which treat all NGD scores uniformly.

Our experiments show that set-based generalization may not provide adequate protection for transaction data, and up to 70% of the items added to transactions during generalization can be detected by our methods with a precision over 80%. Note that in contrast to other studies on quantifying privacy risk involved in publishing transaction data, where adversaries are assumed to either attack de-identified (not generalized) transactions [Narayanan and Shmatikov 2008; Datta et al. 2012] or have some auxiliary information to help recover original data [Giannella et al. 2013], our methods attack anonymized data without using or assuming any background knowledge (e.g. knowing m items about an individual [Terrovitis et al. 2008] or certain association between data items [Martin et al. 2007]) and uses only the information available from the released data. This is significant as it represents a realistic assessment of privacy risk associated with set-based generalization.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we introduce our semantic attack which consists of two main components: *Scoring* and *Elimination*. We describe the scoring component in Section 4, and focus on the elimination component in Section 5. Section 6 reports experimental results, and Section 7 concludes the paper.

2. RELATED WORK

Earlier work on assessing privacy risk associated with data publishing focused on data distribution. For example, link attack [Sweeney 2002] attempts to identify an individual from a dataset by using combinations of certain attribute values (called quasi-identifiers) that do not appear frequently enough; homogeneity attack [Machanavajjhala et al. 2007] aims to reveal sensitive information associated with individuals by considering data distribution within an equivalence group without needing to identify the individuals first; and probabilistic attack [Li et al. 2007] compares data distribution before and after data sanitization, thereby disclosing sensitive information about individuals if probabilities of estimating sensitive information for individuals change significantly after data sanitization. These attacks are syntactic and they do not consider data semantics as we do in this paper.

Later studies considered more advanced attacks using background information beyond the distribution of data. For example, minimality attack [Wong et al. 2007] exploits the fact that data sanitization methods typically seek to minimize data distortion during sanitization, and uses this knowledge to identify how original data may be altered in order to satisfy some minimal protection requirements. Xiao et al. [Xiao et al. 2010] proposed transparency attack which assumes that the adversary knows the algorithm used to sanitize the data, and uses this knowledge to recover original data. Martin et al. [Martin et al. 2007] considered attacks where an adversary has some knowledge about individuals that is not represented in the data, and they capture this knowledge as implications between non-sensitive and sensitive data items and use them when attacking sanitized data. In contrast, our methods do not assume any background knowledge and assess privacy risk using published data only.

There are works that do not directly attempt to identify individuals or the sensitive information associated with them from published datasets, but consider how original data may be recovered from sanitized ones. For example, when a set of data is transformed to protect privacy but also retains Euclidean distances, one may attempt to break the transformation and recover the original values by using a few known data points in the dataset [Giannella et al. 2013]. Narayanan and Shmatikov proposed a method for attacking de-identified transactions [Narayanan and Shmatikov 2008], by using a few known items about an individual to determine other items associated with the individual through some similarity and ranking calculations. Our work shares with these methods in principle: we also attempt to use the published data to break the cover. However, our approach is different from theirs in that we rely on semantic relationships that exist within the dataset, rather than assuming certain data properties, such as Euclidean distances, sparsity of data or known data points.

More recently, Sánchez et al. considered the effect of semantic relationships among data items when sanitizing them [Sánchez et al. 2013]. Their method deals with text data specifically and measures the semantic relationship between two terms. Similar to our work, they also adopt NGD to detect related terms. However, they attempt to establish whether the items remaining in a text after sanitization could be used to recover those that have been suppressed during sanitization. In contrast, our work targets transaction data, and uses items as contextual information to assess if items added by set-based generalization may be removed. Ong and Shao [Ong and Shao 2014] proposed de-anonymization methods to reconstruct original transactions from set-generalized versions through semantic analysis, and the difference between their methods and those proposed here has already been analyzed in the Introduction.

3. SEMANTIC ATTACK EXPLOITING CONTEXTUAL INFORMATION

In this section, we introduce our approach to attacking set-generalized transactions. We first present some notations and concepts necessary to understanding our approach, then give an overview of our approach.

3.1. Preliminaries

Let $\mathcal{I} = \{i_1, ..., i_m\}$ be a finite set of literals called *items*. A *transaction* T over $\mathcal{I}$ is a set of items $T = \langle a_1, a_2, \ldots, a_k \rangle$, where each $a_j, 1 \leq j \leq k$, is a distinct item in $\mathcal{I}$. A transaction dataset $\mathcal{D} = \{T_1, \ldots, T_n\}$ is a set of transactions over $\mathcal{I}$.

Definition 3.1 (*Itemset and Support*). Any subset $I \subseteq \mathcal{I}$ is called an *itemset*. An itemset I is *supported* by transaction T if $I \subseteq T$. We use $\sigma_{\mathcal{D}}(I)$ to represent the number of transactions in $\mathcal{D}$ that support I.

For example, $\langle gangrene, limbs, injury \rangle$ is a transaction in Fig. 1. $\langle limbs, injury \rangle$ is an itemset and is supported by T3, i.e. $\sigma_{\mathcal{D}}(\langle limbs, injury \rangle) = 1$. When the support for an itemset is low, an attacker may use it to identify an individual with a high probability. To address this, various privacy models have been proposed [Terrovitis et al. 2008; Loukides et al. 2011]. For the purpose of this paper, we use a simple, but commonly adopted privacy model based on support count.

Definition 3.2 (*Transaction Protection*). Let $\mathcal{D} = \{T_1, T_2, \ldots, T_n\}$ be a set of transactions, and $p = (I, \sigma_{min})$ be a privacy constraint that requires an itemset I to have a minimum support σ_{min} in $\mathcal{D}$. $\mathcal{D}$ is *protected* w.r.t. p if either $\sigma_{\mathcal{D}}(I) \geq \sigma_{min}$ or $\sigma_{\mathcal{D}}(I) = 0$.

When specified privacy constraints are not satisfied, data must be sanitized. One approach is set-based generalization which replaces individual items by a set of items [Loukides et al. 2011].

Definition 3.3 (*Set-Based Generalization*). A set-based generalization is a partition $\tilde{\mathcal{I}}$ of $\mathcal{I}$ in which each item $i \in \mathcal{I}$ is replaced by the partition to which it belongs. Each partition is called a *generalized item*. When an item is generalized to itself, we say that the item is *trivially generalized*.

We denote a generalized item by listing its items in brackets, e.g. (*blood pressure, icd, limbs, injury*) in Fig. 2, and we interpret it as representing any non-empty subset of its members, e.g. (*blood pressure, icd, limbs, injury*) may represent *blood pressure*, *icd* , *limbs*, *injury* or any combination of them. We call an item in a generalized item *original item* if it is in the original transaction, and an *added item* if it is added to the transaction during generalization. For example, in T4 of Fig. 2, *injury* in (*blood pressure, icd, limbs, injury*) is an original item and *limbs* is an added item.

3.2. Semantic Attack

Given a set of set-generalized transactions, we are interested to see if any added items may be identified from a generalized item by considering the semantic relationships that exist among the items.

Definition 3.4 (*Context*). Given a generalized transaction $\tilde{T}$, C is a *context* of $\tilde{T}$ if C contains only trivially generalized items in $\tilde{T}$. We call an item in C a *contextual item*.

Definition 3.5 (*Semantic Relationship*). Let C be a context of $\tilde{T}$, $\tilde{i}$ be a generalised item in $\tilde{T}$, and $\hat{i}$ be an item in $\tilde{i}$. *Semantic relationship* between $\hat{i}$ and C, denoted by $s(C, \hat{i})$, is a measure of their expected co-occurrence in $\tilde{T}$.

When $\hat{i}$ is deemed unlikely to occur in $\tilde{T}$ with C, we may eliminate $\hat{i}$ from $\tilde{i}$, thereby breaking the generalization. So our semantic attack is based on a form of *item co-occurrence* analysis, which is similar in principle to latent semantic analysis (LSA) in document indexing and retrieval [Deeerwester et al. 1990]. Different approaches, such as machine learning or ontological analysis [Nenkova and McKeown 2012], may be used to derive $s(C, \hat{i})$, and in this paper we adopt Normalized Google Distance (NGD) [Cilibrasi and Vitányi 2007]:

$$NGD(x, y) = \frac{max(log(f(x)), log(f(y))) - log(f(x, y))}{log(N) - min(log(f(x)), log(f(y)))} \quad (1)$$

where $f(x)$ is the number of Google pages containing x, $f(y)$ the number of pages containing y, $f(x, y)$ the number of pages containing both x and y, and N the size of Google repository. The smaller the $NGD(x, y)$ is, the more closely x and y are deemed to be related. For example, we have $NGD(paracetamol, HIV) > NGD(paracetamol, cold)$, suggesting that *paracetamol* is more closely associated with *cold* than with *HIV*.

We adopt NGD in our work for two reasons. First, NGD measures semantic relationship between two items in terms of their co-occurrence in a document, rather than based on their ontological similarity. This is important as the ability to test if two terms can be expected to occur together, but may not necessarily be related ontologically, can help detect added items in a generalized item, as our example above shows. Second, NGD uses Google repository for testing term co-occurrence. The extensive range of topics covered by Google means that term co-occurrence can be tested in many contexts or domains, without needing a comprehensive corpus like LSA requires. This is desirable and makes our methods generic and practical.

3.3. Overview of Our Approach

We observe that when an item in a transaction is generalized, items added to the generalized item may not always be consistent (or is unlikely to occur) with the contextual items surrounding them. Our idea is to use contextual items to detect if any item in a generalized item is an added one. We do so in two steps:

— *Scoring* is a step to establish the strength of relationships between items in a generalized item and contextual items using NGD as a measure. Given a transaction, it is reasonable to expect its items collectively forming some context(s), as the transactions in Fig. 1 demonstrate. Thus, when an item in a generalized transaction displays a weak relationship with contextual items, we can reasonably suspect that it may be an added item. The scoring step is to identify such candidates, by selecting suitable contexts and deriving NGD scores.
— *Elimination* is to determine which candidate items are added items, based on the scores obtained from the scoring step. Note that NGD scores are relative: they indicate if an item is more likely to be related to one item than another, but do not give an absolute verdict on whether two items are definitely related or not. Thus, heuristics are needed. In this paper we develop two such heuristics.

It is worth noting that while we consider semantic attack on set-generalized transactions only in this paper, our approach is applicable to other types of data and sanitisation too. For example, when bucketization is used to protect relational data (even with substantial background knowledge taken into account [Martin et al. 2007]), and when privacy-preserving document indexing [Bawa et al. 2003; Tang et al. 2014; Tang and Liu 2015] is exercised or disassociation is used for transaction sanitization [Terrovitis et al. 2012], only the frequency of data association is altered and semantic relationship among the data items remains unchanged. As such, the sanitized data is still vulnerable to the type of attack we study here. So our approach complements existing privacy assessment by considering semantic relationship rather than frequency of data association. In the next two sections we explain the two steps of our approach in detail.

4. SCORING ITEM RELATIONSHIPS

In this section we consider how a context may be obtained from a generalized transaction and how the relationship between the items in a generalized item and the contextual items may be scored.

4.1. Forming a Context

Given a context $C = \{i_1, \ldots, i_w\}$ and a generalized item $\tilde{i} = (\hat{i_1}, \ldots, \hat{i}_h)$, there are two possible ways to interpret the relationship between C and $\tilde{i}$:

— C is considered as a single "conjunctive" item and $NGD(C, \hat{i}_j)$ is measured. That is, we assess if an item in $\tilde{i}$ is likely to occur with all the items in C.
— C is considered as containing individual items, and the relationship between an item in $\tilde{i}_j$ and each item in C is measured separately and the scores are averaged:

$$d_{C,\hat{i_j}} = \frac{\sum_{c \in C} NGD(c, \hat{i_j})}{|C|} \tag{2}$$

The first approach is more likely to lead to inaccurate estimation when C is large and contains multiple "themes". For example, a transaction extracted from a patient discharge report may contain multiple diseases about the patient, and items in $\tilde{i}_j$ may not necessarily be related to all contextual items. We therefore adopt the second approach.

Note that in forming a context we assume that an adversary knows which item is generalized. This is commonly assumed as making generalization explicit helps enhance data utility. If data are released without generalized items being explicitly marked, for example, brackets are not used to mark generalized items in Fig. 2, then an additional step is needed to identify contextual items first. Further discussion on this is beyond the scope of current paper. Furthermore, our technique assumes that co-occurrence based semantic relationship among items is meaningful. This may not be true for some datasets, for example, in a shopping basket dataset where each transaction records items that a customer has purchased. Such datasets may not have the type of contextual items that we consider in our work. For example, given $\langle$*milk*, (*bread*, *bacon*), *cheese*, *medicine*$\rangle$, it may be inferred that *bread* is most likely to be an original item since it occurs with *milk* and *cheese* more frequently. However, this is unjustified as items in this case are related by shopping preferences, rather than by certain context. Our methods are unsuitable to use to attack this type of transaction data.

4.2. Distance Table

Set-based generalization requires that generalized items form k-equivalence groups. That is, each generalized item will appear at least k times within the released transactions. This is to ensure that the probability of using generalized items to link an individual to a transaction is no more than $1/k$. We therefore consider the whole equivalence group together when attacking a generalized item $\tilde{i} = (\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_s)$, by performing NGD on each occurrence of $\tilde{i}$ in different transactions and record the result in a distance table, as shown in Fig. 3, where columns are items in the generalized item, and rows are contextual items from each transaction in the group. Note that while the generalized item $\tilde{i}$ is identical in every transaction within the equivalence group, the contextual items that are selected to attack it need not be the same. In fact, as each transaction is different, contexts are likely to be different, thereby allowing the membership of $\hat{i}$ in $\tilde{i}$ to be discriminated in a given transaction.

	$\hat{i}_1$	...	$\hat{i}_s$
C_1	$d_{C_1,\hat{i}_1}$	...	$d_{C_1,\hat{i}_s}$
...	...	...	...
C_k	$d_{C_k,\hat{i}_1}$	...	$d_{C_k,\hat{i}_s}$

Fig. 3: Distance Table

To illustrate our method, consider Fig. 2 again. Applying our scoring function to the generalized item (*blood pressure, icd, limbs, injury*), we obtain the distance table in Fig. 4 (we use bold for distances of original items). This generalized item contains 4 items and forms a 4-equivalence group, hence a 4 by 4 distance table. The largest distance is 2.93 between *icd* and *gangrene*, suggesting that they are not perhaps as related as others are, and *icd* is likely to be an item added to T3 by the generalization process. Note that in this example, we used a single contextual item to attack the generalized item. In general, any number of contextual items may be used.

5. ELIMINATING ADDED ITEMS

In our previous work we proposed methods that heuristically eliminate added items based on their NGD scores [Ong and Shao 2014]. These methods treat all NGD scores within a distance table uniformly. For example, *icd* and *gangrene* have the greatest distance in Fig. 4, thus *icd* in T3 is considered most likely to be an added item. This

	blood pressure	icd	limbs	injury
$C_1 = \{\text{heart disease}\}$	**0.56**	**0.78**	1.57	2.19
$C_2 = \{\text{anesthesia}\}$	1.75	**0.58**	1.74	1.53
$C_3 = \{\text{gangrene}\}$	2.60	2.93	**1.78**	**1.49**
$C_4 = \{\text{knee}\}$	1.60	1.51	1.89	**1.03**

Fig. 4: An Example Distance Table

however can lead to wrong eliminations, as we will explain below. In this paper, we propose two clustering-based methods which attempt to identify groups of "comparable distances" first and then apply heuristics to eliminate added items within the groups, rather than treating all distances in a distance table uniformly.

5.1. Grouping-Based Attack (GBA)

NGD measures the relatedness of two terms by their co-occurrences in WWW pages. If an item is not commonly used, its NGD with any other items will be large, even with those they are closely related to. For example, distances associated with $C3$ in Fig. 4 are generally greater than others, because *gangrene* is less commonly used in WWW pages. This implies that treating all values in a distance table uniformly will be biased towards eliminating items in a row or column that contains a rare term. Furthermore, NGD is a relative measure. Given NGD of two pairs of terms $NGD(A, B) = d_1$ and $NGD(C, D) = d_2$, $d_1 < d_2$ does not necessarily mean that A is more related to B than C is to D. However, if we have $NGD(A, C) = d_3$ and $d_1 < d_3$, then it is more reliable to consider that A is more related to B than is to C because we have a common term A for comparison. To address these issues, we introduce the concepts of *comparable distances* and their *vulnerability*.

Definition 5.1 (*Comparable Distances*). Let $d_1 = NGD(A_1, B_1), d_2 = NGD(A_2, B_2), \ldots, d_m = (A_m, B_m)$ be a set of NGD distances. $d_1, d_2, \ldots, d_m$ are comparable if $\bigcap_{i=1}^{m}(\{A_i \cup B_i\}) \neq \emptyset$, where A_i and $B_i, 1 \leq i \leq m$, are sets of items.

Comparable distances are those that involve at least one common item in their NGD assessment, so they are more reliable for comparison. For example, the first column of Fig. 4 form a group of comparable distances as they all involve *blood pressure* in their NGD assessment, and as such $NGD(gangrene, blood\ pressure) = 2.60$ and $NGD(heart\ disease, blood\ pressure) = 0.56$ are comparable, whereas $NGD(gangrene, blood\ pressure) = 2.60$ and $NGD(knee, injury) = 1.03$ are not.

Definition 5.2 (*Vulnerability*). Given a group of comparable distances $\mathbb{C} = \{d_1, d_2, \ldots, d_k\}$ such that $d_1 \leq d_2 \leq \cdots \leq d_k$, its vulnerability is given by

$$\mathcal{V}_{\mathbb{C}} = max(d_2 - d_1, d_3 - d_2, \cdots, d_k - d_{k-1}) \tag{3}$$

Vulnerability of a group of comparable distances is the largest distance gap between two neighbouring items. This gap effectively separates the distances within the group into two clusters, one containing *lower* distances, and the other *higher*. A larger gap between the two clusters suggests that items related and not-related to contextual items within the group are better separated. The higher the vulnerability of a group is, the more likely the higher cluster of the group will contain an added item, and attacking such a group first is more likely to lead to correct eliminations. For example, the vulnerabilities of the second row and third row in Fig. 4 are 0.95 and 0.82, respectively, suggesting that the second row is more likely to contain an added item even though the distances in the third row are greater. Thus, vulnerability allows distances to be compared more meaningfully across groups, especially when rare terms are present.

GBA is based on comparable distances. It treats each row or column of a distance table as a group of comparable distances, and attacks the groups based on their vulnerability. Like our previous methods [Ong and Shao 2014], GBA is an iterative method and the distances are weighted to reflect how likely an item is original. GBA will eliminate one item from the most vulnerable group in each iteration and then update the weighted distances as follows:

Definition 5.3 (*Weighted Distance*). Let $\mathbb{D}$ be a distance table and α_{ij} be the distance value at row i and column j in $\mathbb{D}$. The *weighted distance* α_{ij}^w for α_{ij} is

$$\alpha_{ij}^w = \alpha_{ij} \times (1 - \frac{1}{N_r - E_r^i}) \times (1 - \frac{1}{N_c - E_c^j})$$

where N_r and N_c are the number of rows and columns in $\mathbb{D}$, and E_r^i and E_c^j are the number of eliminated items in row i and column j, respectively.

So if a row (or column) contains m items, then each item is initially assumed to have a probability of $1/m$ to be an original one. As items are eliminated from the distance table, these probabilities will change and we use these probabilities as weights to revise the distances recorded in the table according to Definition 5.3. Details of GBA are given in Algorithm 1.

ALGORITHM 1: *GBA* ($\mathbb{D}, N^r, N^c$)

Input: A distance table $\mathbb{D}$ with N^r rows and N^c columns
Output: $\mathbb{D}$ with added items eliminated

1: $E^c, E^r \leftarrow initialise()$
2: $\mathbb{D}^w \leftarrow weighting(\mathbb{D}, N^r, N^c, E^r, E^c)$
3: $\delta \leftarrow \frac{1}{N^r+N^c} \sum_{D \in \mathbb{D}^w} \mathcal{V}(D)$
4: $v_k \leftarrow max_{D_k \in \mathbb{D}^w} \mathcal{V}(D_k)$ if $d_{ij} = max(D_k), N^r - E_i^r \geq 2, N^c - E_j^c \geq 2$
5: **while** $v_k > \delta$ **do**
6: $\quad \mathbb{D} \leftarrow eliminate(\mathbb{D}, d_{ij})$
7: $\quad E_i^r \leftarrow E_i^r + 1, E_j^c \leftarrow E_j^c + 1$
8: $\quad \mathbb{D}^w \leftarrow weighting(\mathbb{D}, N^r, N^c, E^r, E^c)$
9: $\quad v_k \leftarrow max_{D_k \in \mathbb{D}^w} \mathcal{V}(D_k)$ if $d_{ij} = max(D_k), N^r - E_i^r \geq 2, N^c - E_j^c \geq 2$
10: **end while**
11: **return** $\mathbb{D}$

Step 1 initializes E_r and E_c which are used to keep tracking the number of eliminations in each row and column respectively. This helps update the weighted distance table more efficiently. Step 2 calculates the initial weighted distance table, and a vulnerability threshold δ is calculated based on the initial $\mathbb{D}^w$ in Step 3, which is the average vulnerability of all groups of comparable distances within the table. Note that this average is not re-calculated during iterations as the threshold calculated from all distances is more meaningful and reliable to use. Step 4 selects the group with the greatest vulnerability according to Definition 5.2, as long as the group satisfies the following condition: the item with the largest value (d_{ij}) in this group is not the last item in the corresponding row or column. The attacking criteria is checked in Step 5 to see if the vulnerability of the group is above the threshold. If it is, d_{ij} is removed from the distance table in Step 6. E_c and E_r are then updated and $\mathbb{D}^w$ and v_k are re-calculated based on the new $\mathbb{D}$ in Steps 8 and 9. The process terminates when the attacking criteria is no longer satisfied and the result is returned in Step 11.

Applying GBA to the distance table in Fig. 4 gives the results in Fig. 5 after the first iteration, where the group for column *icd* is selected because it has the highest

vulnerability of 0.80 and is above the threshold of 0.45, and *icd* is eliminated from T3. Following the elimination, the weight table is updated as shown in Fig. 5(a).

	bp	icd	limbs	injury
C_1	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$
C_2	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$
C_3	$\frac{1}{3} : \frac{1}{4}$	- : -	$\frac{1}{3} : \frac{1}{4}$	$\frac{1}{3} : \frac{1}{4}$
C_4	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$

(a) Row:Column Weights

	bp	icd	limbs	injury
C_1	**0.32**	**0.39**	0.88	1.23
C_2	0.98	**0.29**	0.98	0.86
C_3	1.30	-	**0.89**	**0.75**
C_4	0.90	0.76	1.06	**0.58**

(b) Weighted Table

Fig. 5: The First Iteration of GBA

It is worth observing the calculation of group vulnerability and its effect on the elimination process at this point. First, group vulnerability is re-calculated after each iteration, and it may go up or down following an elimination. However, as each iteration involves a different set of distances with different weights applied, group vulnerabilities should not be compared across different iterations, as such comparisons are not meaningful. Second, eliminating one item from one group will affect not only the vulnerability of this group, but the vulnerabilities of other groups too. For example, when *icd* is eliminated from T3 (third row) in Fig. 5, all the distances in the *icd* column are affected through weight distribution. This in turn can change, for instance, the vulnerability of T1 (first row). As such, group vulnerability helps target items to eliminate iteratively, albeit in a greedy fashion.

Continuing the process, at the fifth iteration (Fig. 6), the highest vulnerability is 0.41 (with the *blood pressure* column) $<$ the threshold, so the process terminates.

	bp	icd	limbs	injury
C_1	$\frac{1}{3} : \frac{1}{2}$	$\frac{1}{3} : \frac{1}{3}$	$\frac{1}{3} : \frac{1}{3}$	- : -
C_2	- : -	$\frac{1}{2} : \frac{1}{3}$	- : -	$\frac{1}{2} : \frac{1}{3}$
C_3	- : -	- : -	$\frac{1}{2} : \frac{1}{3}$	$\frac{1}{2} : \frac{1}{3}$
C_4	$\frac{1}{4} : \frac{1}{2}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{3}$

(a) Row : Column Weights

	bp	icd	limbs	injury
C_1	**0.19**	**0.35**	0.70	-
C_2	-	**0.19**	-	0.51
C_3	-	-	**0.59**	**0.50**
C_4	0.60	0.76	0.95	**0.52**

(b) Weighted Table

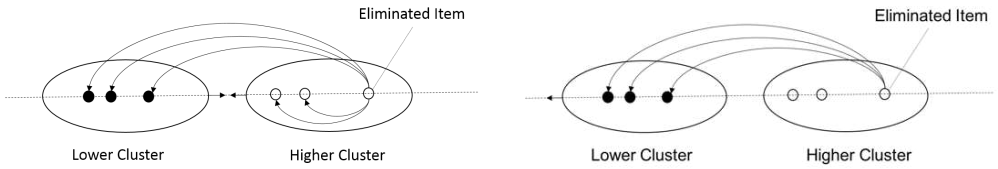
Fig. 6: The Fifth (final) Iteration of GBA

As we will show in Section 6, GBA can identify more added items than our previous methods do. Also it is less affected by the ratio of the number of original items present in generalized items. Therefore, GBA can be expected to have a better trade-off between the precision of elimination and the number of added items that can be identified, than our previous methods do.

5.2. Redistribution-based Attack (RBA)

With GBA, once an item is eliminated, we consider the effect of elimination on the rest of the items within the group to be equal. That is, we redistribute the weight of the eliminated item to the rest of the items equally. This makes all these items equally more likely to be an original item. This however can lead to wrong eliminations, as illustrated in Fig. 7. When the weight of an eliminated item is redistributed to all items in the group, it causes all the distances to be reduced. This in turn causes a) the gap between the lower and higher clusters to be reduced, and b) shifts the distances in the higher cluster towards the mean, as Fig. 7 (a) illustrates. As the threshold (the

mean) calculated at the start of the process is fixed, this makes the remaining added items harder to detect as the iteration progresses. On the other hand, if the weight of an eliminated item is redistributed to the items in the lower cluster, then only the distances in this cluster will decrease, as illustrated in Fig. 7 (b). This will not prevent those in the higher cluster to be considered for elimination, as their distances to the mean are not affected.



(a) Redistributing weight equally to all items (b) Redistributing weight equally to lower items

Fig. 7: Different redistributions of weights after eliminating an item

In this section, we propose RBA which distributes the weight of an eliminated item to the items in the lower cluster equally. The method is similar to GBA, except that the *weighting* function used in Algorithm 1 is replaced by the *RWeighting* function given in Algorithm 2, where π^r and π^c are row and column projection functions, respectively, and the $lower$ function returns the items in the lower cluster of a group.

ALGORITHM 2: *RWeighting* $(\mathbb{D}, W^r, W^c, i, j)$

Input: A distance table $\mathbb{D}$, row and column weight tables W^r, W^c and indices i and j of the item eliminated
Output: Updated weight tables W^r and W^c
1: $D^r \leftarrow \pi_i^r(\mathbb{D})$, $D^c \leftarrow \pi_j^c(\mathbb{D})$
2: **for** $d_y \in lower(D^r)$ **do**
3: $\quad W^r_{iy} \leftarrow W^r_{iy} + \frac{w^r_{ij}}{|lower(D^r)|}$
4: **end for**
5: **for** $d_x \in lower(D^c)$ **do**
6: $\quad W^c_{xj} \leftarrow W^c_{xj} + \frac{w^c_{ij}}{|lower(D^c)|}$
7: **end for**
8: **return** W^r, W^c

Step 1 obtains two groups, D^r and D^c, from the distance table, which are the row and column that contain the eliminated item d_{ij}. Step 2 loops over each item d_y in the lower cluster of D^r, and its corresponding weight in the row weight table is adjusted in Step 3 by adding $\frac{w^r_{ij}}{|lower(D^r)|}$ to the current weight, where w^r_{ij} is the weight of the eliminated item in the row weight table and $|lower(D^r)|$ is the number of items in the lower cluster. That is, the weight of the eliminated item is divided equally among the items in the lower cluster. The column weight table is updated similarly in Steps 5 and 6. Finally, Step 7 returns the two updated weight tables.

Consider the example we used to illustrate GBA again. Applying RBA to the distance table gives the same result in the first iteration and *icd* in T3 is eliminated as it has the greatest distance value in the group and the group has the greatest vulnerability. However, the weight of *icd* is distributed differently this time. In row *C3* in Fig. 8 (a),

the weight is distributed to *limbs* and *injury* only, and *blood pressure* does not receive any additional weight because it is in the higher cluster. In column *icd* of Fig. 8 (a), all items receive weight because they are all in the lower cluster, and *icd* is the only item in the higher cluster which has been eliminated. A new weighted distance table is given in Fig. 8 (b).

	bp	icd	limbs	injury
C_1	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$
C_2	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$
C_3	$\frac{1}{4} : \frac{1}{4}$	- : -	$\frac{3}{8} : \frac{1}{4}$	$\frac{3}{8} : \frac{1}{4}$
C_4	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{3}$	$\frac{1}{4} : \frac{1}{4}$	$\frac{1}{4} : \frac{1}{4}$

(a) Row : Column Weights

	bp	icd	limbs	injury
C_1	**0.32**	**0.39**	0.88	1.23
C_2	0.98	**0.29**	0.98	0.86
C_3	1.46	-	**0.83**	**0.70**
C_4	0.90	0.76	1.06	**0.58**

(b) Weighted Table

Fig. 8: The First Iteration of RBA

Continuing the process and after ten iterations, there is no group that has a vulnerability higher than the threshold and the method terminates. We obtain Fig. 9 as a result. In this particular example, RBA has achieved a remarkable accuracy due to its weight redistribution mechanism, and our experiments show that RBA generally outperforms GBA and the other methods we proposed previously [Ong and Shao 2014].

	bp	icd	limbs	injury
C_1	$\frac{1}{2} : 1.0$	$\frac{1}{2} : \frac{1}{2}$	- : -	- : -
C_2	- : -	$1.0 : \frac{1}{2}$	- : -	- : -
C_3	- : -	- : -	$\frac{1}{2} : 1.0$	$\frac{1}{2} : \frac{1}{2}$
C_4	- : -	- : -	- : -	$1.0 : \frac{1}{2}$

(a) Row : Column Weights

	bp	icd	limbs	injury
C_1	**0.0**	**0.20**	-	-
C_2	-	**0.0**	-	-
C_3	-	-	**0.0**	**0.37**
C_4	-	-	-	**0.0**

(b) Weighted Table

Fig. 9: The Tenth Iteration of RBA

6. EXPERIMENTS

In this section, we evaluate the proposed methods empirically. We measure the effectiveness of our attacks in terms of recall and precision. That is, the more added items can be eliminated correctly by a method, the more effective the method is.

6.1. Dataset Preparation and Experiment Setup

Our experiments were conducted on the transactions extracted from three real datasets: AOL (search queries)[1], I2B2 (medical texts)[2] and GoArticle (general articles)[3]. Table I summarizes their properties.

— *Density* is the average ratio of the number of original items to the total number of items in a generalized item.
— *Length* is the average number of items in a transaction.
— *Source Format* indicates the type of data from which we obtain the dataset.
— *Quality* indicates if the data contains many typos and abbreviations (T&A).
— *Domain* indicates if the dataset covers a single or multiple domains.

[1] http://gregsadetsky.com/aol-data/
[2] http://i2b2.org
[3] http://www.goarticles.com

Table I: Dataset Properties

	AOL	I2B2	GoArticle
Density	0.22	0.28	0.55
Length	23	216	104
Source Format	Transaction	Text	Text
Quality	Many T&A	Many T&A	Few T&A
Domain	Multiple	Healthcare	Multiple

All these properties can affect the performance of our methods, as we shall see later, hence they provide a good testbed for studying our methods. In the following, we describe the preparation of these datasets in detail.

6.1.1. I2B2. I2B2 has 630 text documents containing de-identified clinical data and patient discharge reports. We choose this dataset because healthcare is one of the most relevant areas for privacy protection. We take the following steps to transform the text data into transactions:

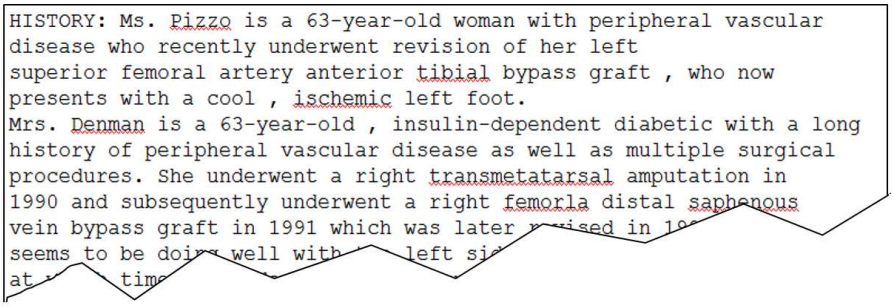
— The Stanford's Part-Of-Speech Tagger[4] is used to extract nouns and noun phrases (i.e. noun + noun, adjective + noun or noun) from a text to form a transaction.
— Some parts of a text, such as headers and footers of a report, are ignored as they are repetitive and not useful in analytic studies.
— We remove any item that (1) is a stop word; (2) is duplicated including singular and plural forms; (3) is contained in another item (e.g. removing "history" if we have "long history" already), because a noun phrase is generally considered to be more meaningful than a noun.

Fig. 10a shows an example document of the dataset and Fig. 10b shows the extracted transaction. The documents in the dataset contain many typographical errors and abbreviations which can affect the results. We retained abbreviations because many common abbreviations can be understood by the Google Search Engine, but we do not attack generalized items that contain typos. Note that although we do not distinguish singular and plural terms, it does not affect the semantic measurement when using NGD because Google automatically searches for both cases.

6.1.2. AOL. The AOL dataset contains about 20M search queries from 650k users. Each record has user_id, timestamp, one or more search keywords, the url clicked and the clicked rank. AOL data is already in a transaction form, but we reformat the data to remove unnecessary information. Fig. 11 (a) shows the data in its original form and Fig. 11 (b) shows the transactions obtained following these preparation steps:

— We first divide the dataset into transactions by user sessions. That is, queries posed with the same user_id (AnonID in Fig. 11 (a)) are put into one transaction. We remove user_id, timestamp and clicked url from the dataset. A keyword may be searched multiple times, and we only include it once.
— We then extract nouns and noun phrases as we did for I2B2. However, when search queries are short containing one or two terms, it is difficult to process them using the normal part of speech tags as they do not follow proper grammar. So we consider the whole query as a single item in such cases. For longer search queries, we use the POS tagger to extract them.
— Similar to the I2B2 dataset, we also apply the rules to remove stop words or redundant words.

[4]http://nlp.stanford.edu/software/tagger.shtml


HISTORY: Ms. Pizzo is a 63-year-old woman with peripheral vascular disease who recently underwent revision of her left superior femoral artery anterior tibial bypass graft , who now presents with a cool , ischemic left foot.
Mrs. Denman is a 63-year-old , insulin-dependent diabetic with a long history of peripheral vascular disease as well as multiple surgical procedures. She underwent a right transmetatarsal amputation in 1990 and subsequently underwent a right femorla distal saphenous vein bypass graft in 1991 which was later revised in 19
seems to be doing well with left si
at time

(a) A sample of I2B2 text

⟨*ms. pizzo, 63-year-old woman, vascular disease, revision, femoral artery, tibial bypass, graft, left foot, mrs. denman, long history, surgical procedure, transmetatarsal, amputation, right femorla, saphenous vein, by pass graft,* · · · · · ·⟩

(b) A sample of I2B2 extracted transaction

Fig. 10: An example of I2B2 text and extracted transaction

1	AnonID	Query	QueryTime	ItemRank	ClickURL
8303	13508	good wine	3/1/2006 21:29	7	http://www.wines.com
8304	13508	accent marks	3/1/2006 22:04	1	http://faculty.weber.edu
8305	13508	accent marks	3/1/2006 22:04	2	http://fog.ccsf.cc.ca.us
8306	13508	accent marks	3/1/2006 22:04	4	http://users.ipfw.edu
8307	13508	accent marks	3/1/2006 22:04	3	http://www.starr.net
8308	13508	accent marks	3/1/2006 22:04	10	http://en.wikipedia.org
8309	13508	body mass index	3/2/2006 20:34	2	http://www.halls.md
8310	13508	por que te vas	3/2/2006 21:04	1	http://members.fortunecity.es
8311	13508	por que te vas	3/2/2006 21:04	8	http://lyricsplayground.com
8312	13508	sudden weight loss	3/3/2006 19:21	1	http://menshealth.about.com
8313	13508	sudden weight loss	3/3/2006 19:21	3	http://www.ivillage.co.uk
8314	13508	not enough sleep	3/3/2006 19:26	4	http://search400.techtarget.com
8315	13508	not enough sleep	3/3/2006 19:26	3	http://www.webmd.com

(a) A sample of AOL data

⟨*good wine, accent mark, body mass, index, por que, te va, sudden weight, loss, cancer.org, chemotherapy, fertility, methotrexate, love, eid al, fitr, al fitr, crush*⟩

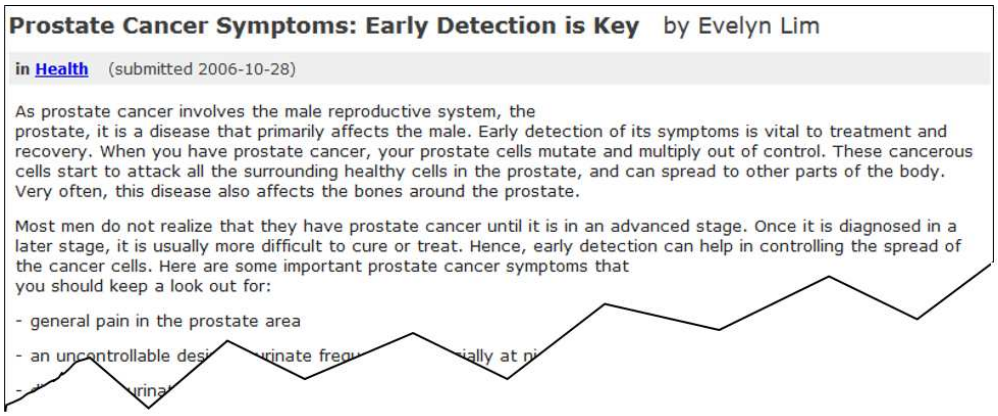
(b) A sample of AOL extracted transaction

Fig. 11: An example of AOL data and extracted transaction

While I2B2 is about the medical domain only, search queries in the AOL dataset are about various topics. Therefore, when generalizing them, keywords from different domains may be grouped together. This may make added terms easier to detect, and this dataset will test how our methods work on this type of data.

6.1.3. GoArticle. The AOL and I2B2 datasets have a similar property: their generalized items tend to have a low density. To evaluate our methods on datasets with a higher density, we constructed the GoArticle dataset, which is collected from GoArticles.com on some specific topics. We manually chose articles which share many common keywords, forming relatively dense transactions.

GoArticle contains free text which is similar to the I2B2 dataset. So the same process is followed to prepare and extract transactions from free text. Fig. 12 (a) shows a sample document and Fig. 12 (b) shows an extracted transaction.



Prostate Cancer Symptoms: Early Detection is Key by Evelyn Lim

in Health (submitted 2006-10-28)

As prostate cancer involves the male reproductive system, the prostate, it is a disease that primarily affects the male. Early detection of its symptoms is vital to treatment and recovery. When you have prostate cancer, your prostate cells mutate and multiply out of control. These cancerous cells start to attack all the surrounding healthy cells in the prostate, and can spread to other parts of the body. Very often, this disease also affects the bones around the prostate.

Most men do not realize that they have prostate cancer until it is in an advanced stage. Once it is diagnosed in a later stage, it is usually more difficult to cure or treat. Hence, early detection can help in controlling the spread of the cancer cells. Here are some important prostate cancer symptoms that you should keep a look out for:

- general pain in the prostate area

- an uncontrollable desi urinate frequ ially at ni

(a) A sample of GoArticle text

⟨*prostate cancer, reproductive system, disease, male, detection, symptom, treatment, recovery, prostate cell, healthy cell, part, body, bone, man, advanced stage, later stage, early detection, spread, cancer cell, important prostate,* $\cdots\cdots$⟩

(b) A sample of GoArticle extracted transaction

Fig. 12: An example of GoArticle text and extracted transaction

6.1.4. Experiment Setup. We used COAT [Loukides et al. 2011] to anonymize a set of transactions. COAT requires the user to specify the following inputs:

— Privacy constraints. These specify which items in a given transaction dataset need to be protected. In our experiments, we randomly select $x\%$ items as privacy constraints.
— Protection parameter k. This parameter ensures that any subset of items in a privacy constraint appears at least k times in the transactions. In our experiments, we varied k from 2 to 6.
— Utility constraints. They specify what can be used to generalize an item in transactions. In our experiments, we use two types of utility constraint. One simply uses all the items as single utility constraint. This may cause significant utility losses, but offers better privacy protection as it creates more "diverse" generalizations. We also use a WordNet based approach to specifying utility constraints.

Table II summarizes the characteristics of the datasets we used in experiments. Extracted transactions are the number of transactions that are constructed from the original data. Generalized items is the number of generalized items to be attacked. This number is relatively small compared with the actual number of generalized items in the dataset, but is sufficient to test our methods.

6.2. Results and Discussion

In this section we report our experimental results. For comparison, we have also included the results from our previous methods [Ong and Shao 2014]. Random Attack

Table II: Characteristics of the datasets used in experiments

Dataset	Extracted transaction	Generalized items
AOL	758	127
I2B2	643	112
GoArticle	263	45

(RA) randomly decides if an item is added and serves as a baseline method for comparison. Maximum Distance Attack (MDA) is a conservative method which eliminates the largest distance from a distance table in attack. Threshold Based Attack (TBA) eliminates all distances that are above a specified threshold, so its effectiveness relies on a good threshold. Weight Based Attack (WBA) is similar to GBA and RBA in that it also involves weights and weight redistribution, but it treats all distances uniformly. The reader is referred to [Ong and Shao 2014] for details of these methods.

6.2.1. Effect of k. Fig. 13 shows the effect of varying k on the recall (how many added items are identified), precision (how many identified items are actually added items) and F-score (combination of recall and precision) of each method.

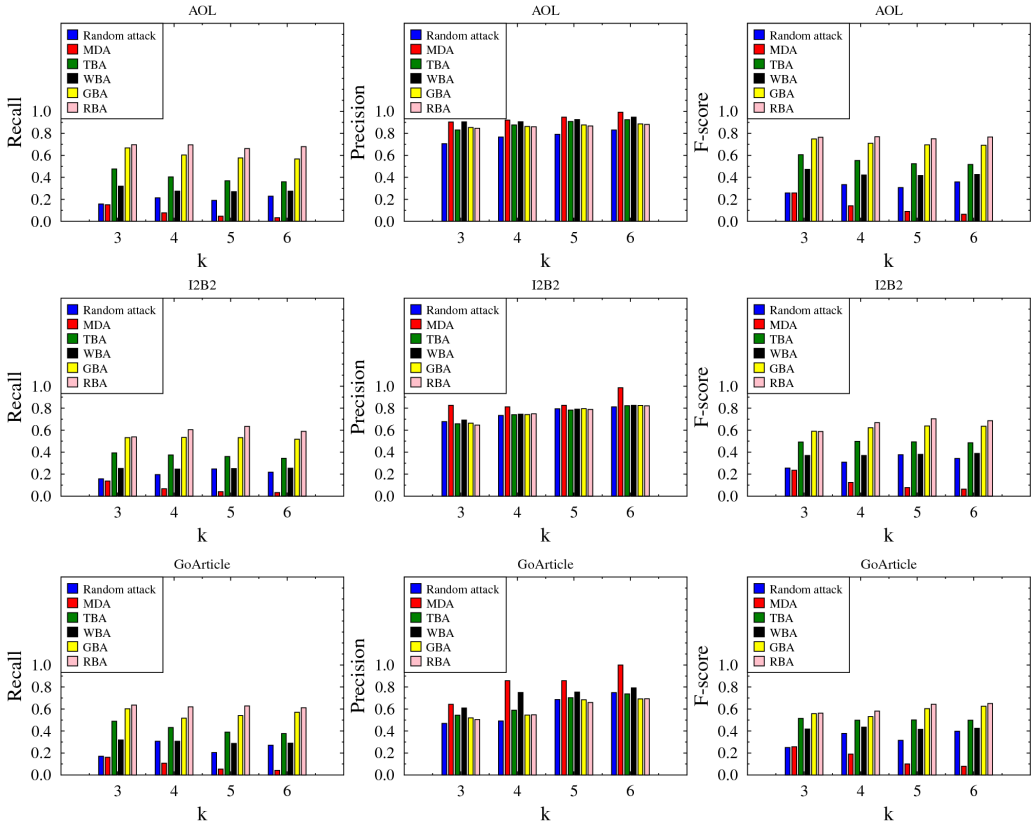


Fig. 13: Recall, Precision and F-Score vs k

In terms of datasets, all the methods performed better on AOL, with RBA achieving an F-score of about 80% for all k values. This is mainly because AOL has a relatively lower density, making it more likely for an added item to be eliminated. This was manifested by the fact that RA has achieved a precision of 70% on AOL but with a very low recall. Furthermore, many AOL transactions contain multiple, distinct contexts, and when items from different contexts mix during generalization there is a better chance to identify added items. This can be seen by comparing AOL to I2B2. I2B2 has a similar density to AOL, but I2B2 items are from the single healthcare domain. This makes added items more difficult to identify, hence lower F-scores. Finally wtih GoArticle, GBA and RBA achieved relatively low precision, although their F-scores are still superior to the other methods. This was the result of clustering: using the largest distance gap to split data into two groups is not effective in some cases.

On the methods, MDA gave low recall and high precision. This is expected as MDA only attempts to eliminate one added item, so is likely to achieve high precision, especially when the density of a dataset is low. Both WBA and TBA can achieve high precision, but can only eliminate a small number of added items. This is because when the density of a dataset is low, the average-based threshold will be high. This leaves many added items below the threshold and not eliminated. GBA and RBA, on the other hand, divide items into comparable groups, focusing on items that are more likely to be added and using early rounds of eliminations to help elimination in the later rounds selectively, thereby improving the overall effectiveness.

6.2.2. Effect of Data Density. We also tested our methods with different density levels. To set up the experiment, we selected subsets of documents (transactions) from the GoArticle dataset to have an average density from 0.1 to 0.7, and the result is shown in Fig. 14. We do not include RA and MDA here as they are not affected by density.

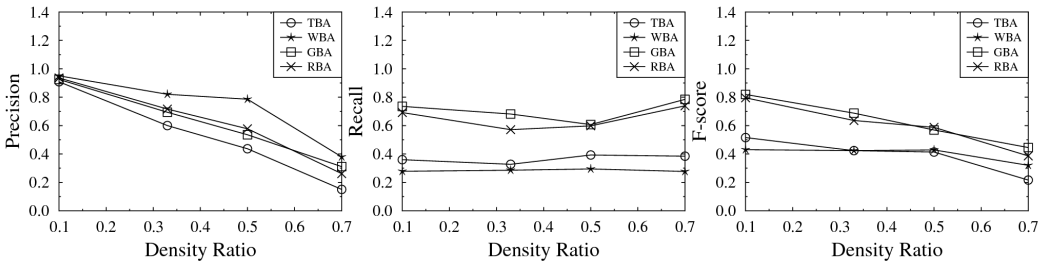


Fig. 14: Recall, Precision and F-Score vs Density

Generally, the precision of the methods decreased when the density increased. This is because when the density is high, many transactions share the same items and fewer items are needed to add into a transaction during generalization. On the other hand, all our methods rely on a notion of average threshold, and when many items in a generalized item are original, the average tends to be below some original items, resulting in some original items to be eliminated. The recall, on the other hand, was not affected by the density, with GBA and RBA showing a strong performance in eliminating a large proportion of added items across all levels of density. This demonstrates the effectiveness of our clustering-based methods: GBA and RBA out-performed the other methods in detecting and eliminating added items.

6.2.3. Effect of Utility Constraints. In the previous experiments, we used the most general utility constraint to anonymize a set of transactions: any item can be used to form a generalizing set. This helps test our methods in general, but one could argue that this makes added items potentially easier to identify, as it is more likely for the generalization process to mix semantically inconsistent items in a single generalized item.

To test how our methods would deal with more carefully constructed utility constraints, we carried out an experiment where a dataset is anonymized by more semantically consistent utility constraints. That is, we only use the items that semantically related to the item to be protected to form a generalising set during generalization. We did this experiment with the GoArticle dataset as its multiple contexts and higher density allow us to construct some semantically very consistent utility constraints.

We used the same setup for the GoArticle dataset as in Fig. 13, except that utility constraints are constructed using the following steps:

— We extract all the items appeared in the dataset to form the domain $\mathcal{I}$ of the dataset. We include all items so that COAT is less likely to suppress an item.
— Given a privacy constraint $(i_1, i_2, \ldots, i_n)$, we search all items in $\mathcal{I}$ which are semantically close to i_1. To determine the closeness of two items i_m and i_n, we use a similarity measure given by Wu and Palmer [Wu and Palmer 1994] based on the WordNet:

$$similarity = 2 \times depth(i_m, i_n)/(depth(i_m) + depth(i_n))$$

where $depth(i_m)$ and $depth(i_n)$ are the distances from the root to items i_m and i_n on the ontology tree, and $depth(i_m, i_n)$ is the distance from the root to the lowest common ancestor of i_m and i_n. Two items are deemed to be sufficiently related if their similarity score is above 0.5.
— We do the same for other items in the privacy constraint. This results in a utility constraint that contains only the items that are semantically consistent with the privacy constraint. Because not all items in the domain will be added into a utility constraint, COAT may need to suppress some items in order to satisfy the privacy constraint. However, as our methods do not consider attacking suppressed items, suppressed items are ignored during attack and are not included in distance tables.

Fig. 15 compares the results of attacking the same dataset anonymized using general utility constaints and semantically consistent ones. For clarity of presentation we only show the results from TBA and RBA here; other methods displayed a similar pattern. We use TBA' and RBA' to denote the results associated with the dataset that are generalized using semantically consistent utility constraints, and TBA and RBA the general ones. As can be seen, recalls associated with semantically consistent utility constraints are slightly lower as a result of introducing semantically more consistent items into transactions, resulting in a distance table with close distances and less items being above the average distance. The precisions are also slightly lower because some added items are closely related to the context, causing some original items to be eliminated. However, the overall results in F-Score show that our methods are not significantly affected by utility constraints, hence we believe that the context of transaction can be used to identify added items even though semantically similar or consistent items are used in generalization.

6.2.4. Efficiency. Fig. 16 shows the efficiency of our methods. The performance of our methods is dependent on the size of distance table, so we evaluate the performance by varying distance table sizes. We use $N^r \times N^r$ to denote the size of a distance table, where N^r is the number of rows and N^c the number of columns. Fig. 16 shows the time taken to process a distance table.

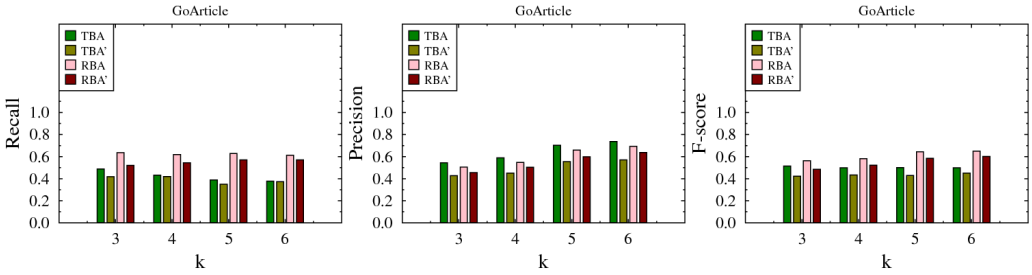


Fig. 15: General vs Semantically Consistent Utility Constraints

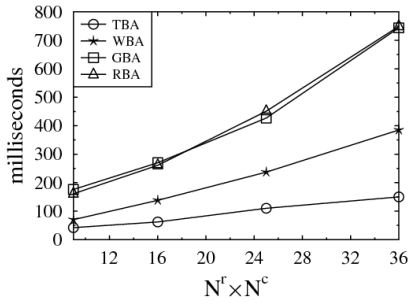


Fig. 16: Efficiency

It is easy to show that TBA has a complexity of $\mathcal{O}(N^r \times N^c)$ and WBA, GBA and RBA have $\mathcal{O}((N^r \times N^c)^2)$. This is confirmed in the experiments, as can be seen in Fig. 16. It can also be seen that although GBA and RBA have a similar complexity to WBA, WBA has a better response time because RBA and GBA need extra time to cluster items. Note that we have not included the time for NGD scoring in this experiment as the process is run remotely on Google servers and is dependent on the Internet speed and external searching algorithms. It is however a slow process: it took more than 24 hours to perform the NGD scoring in our experiments.

7. CONCLUSIONS

In this paper, we have studied if protection for transactions offered by set-based generalization is sufficient. We have proposed methods to attack anonymized data by exploiting contextual information available within the released dataset. Our study has identified a significant issue that has been overlooked by the existing privacy models, and our experiments show that our proposed methods can eliminate up to 70% of added items with a precision about 80%. As our methods do not rely on any background knowledge that an adversary may have, the privacy risks that we identify here are real.

Our work can be extended in a number of directions. First, it is worth investigating a more powerful and accurate scoring approach for semantic attack. Second, we have concentrated on set-based generalization of transaction data. The attacking approach has the potential to be applied to other types of data and privacy models. Finally, it will be useful to study how the confidence of an elimination may be established, which will allow better elimination heuristics to be developed.

REFERENCES

M. Barbaro and T. Zeller. 2006. A face is exposed for AOL searcher no. 4417749. *New York Times* (2006).

M. Bawa, R. J. Bayardo Jr, and R. Agrawal. 2003. Privacy-preserving indexing of documents on the network. In *Proceedings of the 29th International Conference on VLDB*. 922–933.

R.L. Cilibrasi and P.M.B. Vitányi. 2007. The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering* 19, 3 (2007), 370–383.

A. Datta, D. Sharma, and A. Sinha. 2012. Provable de-anonymization of large datasets with sparse dimensions. In *Principles of Security and Trust*. 229–248.

S. Deeerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391 – 407.

C. R. Giannella, K. Liu, and H. Kargupta. 2013. Breaching Euclidean distance-preserving data perturbation using few known inputs. *Data & Knowledge Engineering* 84 (2013), 93–110.

P. Golle. 2006. Revisiting the uniqueness of simple demographics in the US population. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*. 77–80.

N. Li, T. Li, and S. Venkatasubramanian. 2007. t-Closeness : Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering*. 106–115.

G. Loukides, A. Gkoulalas-Divanis, and B. Malin. 2011. COAT: COnstraint-based anonymization of transactions. *Knowledge and Information Systems* 28, 2 (2011), 251–282.

G. Loukides, A. Gkoulalas-Divanis, and J. Shao. 2013. Efficient and flexible anonymization of transaction data. *Knowledge and Information Systems* 36, 1 (2013), 153–210.

A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. 2007. l-Diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007).

D. J. Martin, D. Kifer, A. Machanavajjhala, and J. Gehrke. 2007. Worse-case background knowledge for privacy-preserving data publishing. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE)*.

A. Narayanan and V. Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*. 111–125.

A. Nenkova and K. McKeown. 2012. A Survey of Text Summarization Techniques. In *Mining Text Data*, C. C. Aggarwal and C. Zhai (Eds.). 43–76.

H. Ong and J. Shao. 2014. De-anonymising set-generalised transactions based on semantic relationships. In *Proceedings of the First International Conferenc on Future Data and Security Engineering*. 107–121.

D. Sánchez, M. Batet, and A. Viejo. 2013. Detecting term relationships to improve textual document sanitization. In *Proceedings of Pacific Asia Conference on Information Systems*. 105–119.

L. Sweeney. 2002. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 557–570.

Y. Tang and L. Liu. 2015. Privacy-Preserving Multi-Keyword Search in Information Networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2015), 2424 – 2437.

Y. Tang, L. Liu, A. Iyengar, and K. Lee amd Q. Zhang. 2014. e-PPI: Locator Service in Information Networks with Personalized Privacy Preservation. In *Proceedings of IEEE 34th International Conference on Distributed Computing Systems (ICDCS)*. 186–197.

M. Terrovitis, J. Liagouris, N. Mamoulis, and S. Skiadopoulos. 2012. Privacy Preservation by Disassociation. *Proceedings of the VLDB Endowment (PVLDB)* 5, 10 (2012), 944 – 955.

M. Terrovitis, N. Mamoulis, and P. Kalnis. 2008. Privacy-preserving anonymization of set-valued data. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*. 115–125.

R.C. Wong, A.W. Fu, K. Wang, and J. Pei. 2007. Minimality attack in privacy preserving data publishing. In *Proceedings of the 33rd International Conference on VLDB*. 543–554.

Z. Wu and M. Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. 133–138.

X. Xiao, Y. Tao, and N. Koudas. 2010. Transparent anonymization: Thwarting adversaries who know the algorithm. *ACM Transactions on Database Systems (TODS)* 35, 2 (2010).